High-Speed Network Communication on the Cray XT

Kitrick Sheets
n8851@cray.com

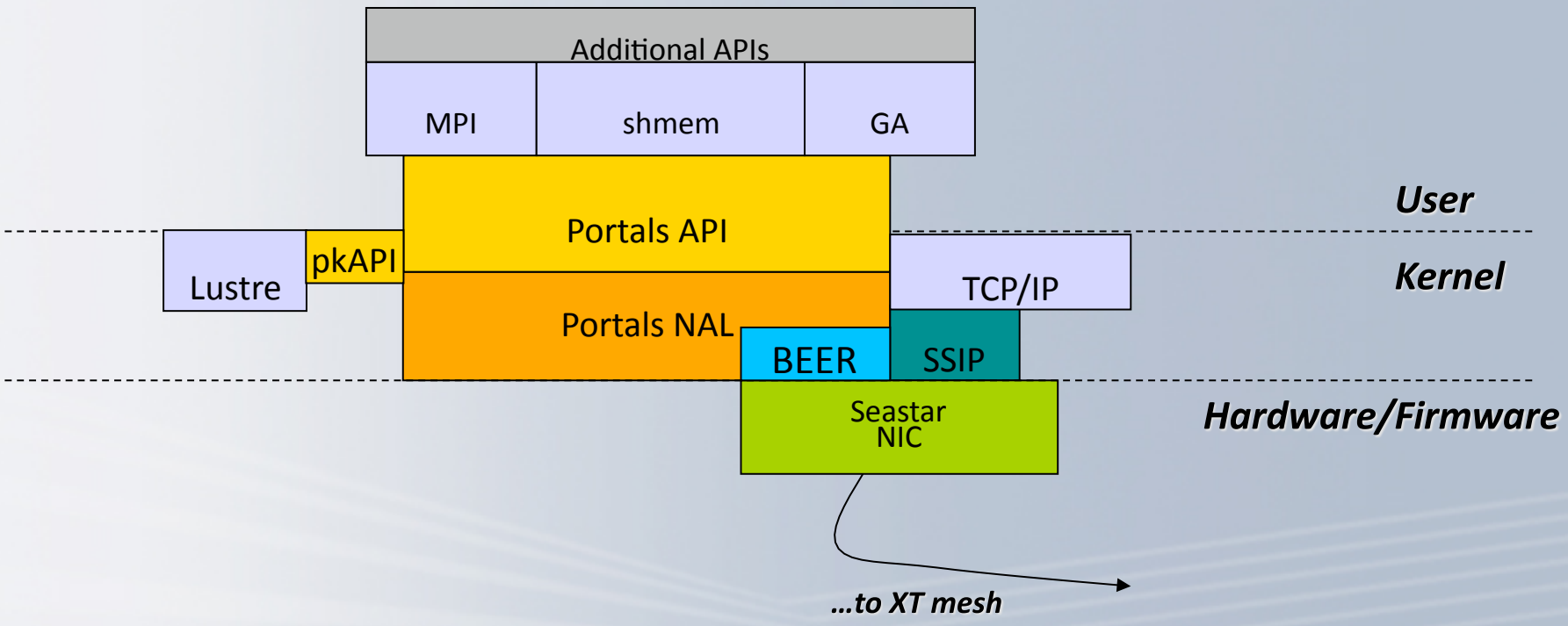
CRAY
THE SUPERCOMPUTER COMPANY

Agenda

- XT communication overview
- Portals overview
- Interconnect specifics
- Interesting communication patterns
- Wrap-up

CRAY
THE SUPERCOMPUTER COMPANY

XT Communication Stack



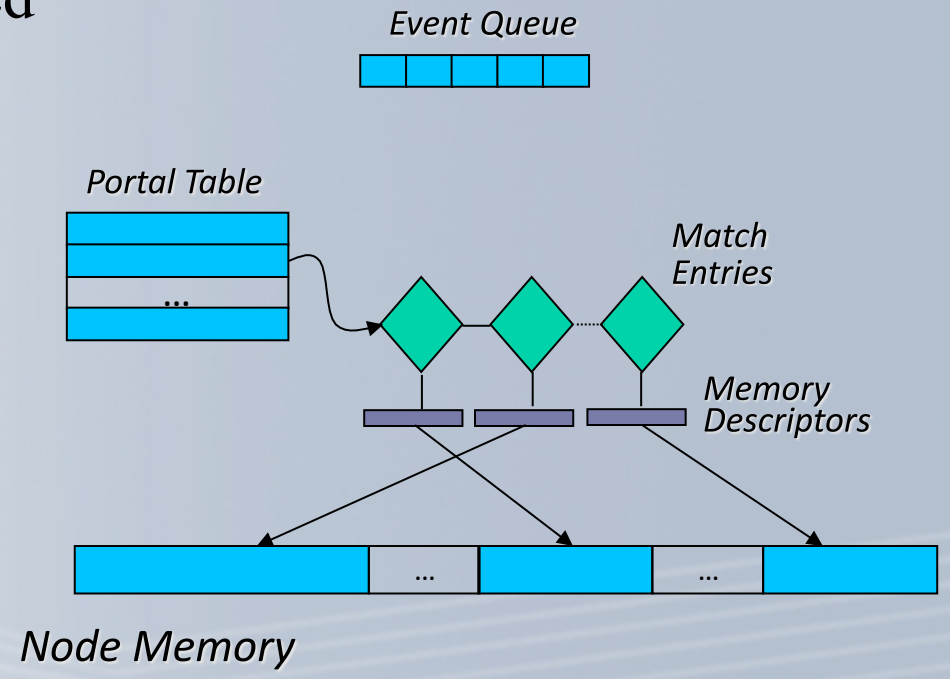
Portals

- Network communication API developed by Sandia
  - Data movement is between ‘portal’s into user address spaces
  - Supports zero-copy, OS bypass, and Application bypass
  - Supports two-way communication (Get and Put)
- Primary method of communication on XT HSN
  - Cray extensions for:
    - CLE optimization
    - Interrupt management improvement
    - Support for multi-core nodes (aprun binding, etc.)
    - Atomic operations (GetPut, GetAdd, CGetPut)
    - Network resiliency (BEER)
    - Scheduling support
    - Scaling to XT5/Petaflop and beyond
- Portals Assumptions
  - Network is reliable and deterministic
  - In-order transmission of messages between NID/PID pairs

CRAY
THE SUPERCOMPUTER COMPANY

Portals Structure

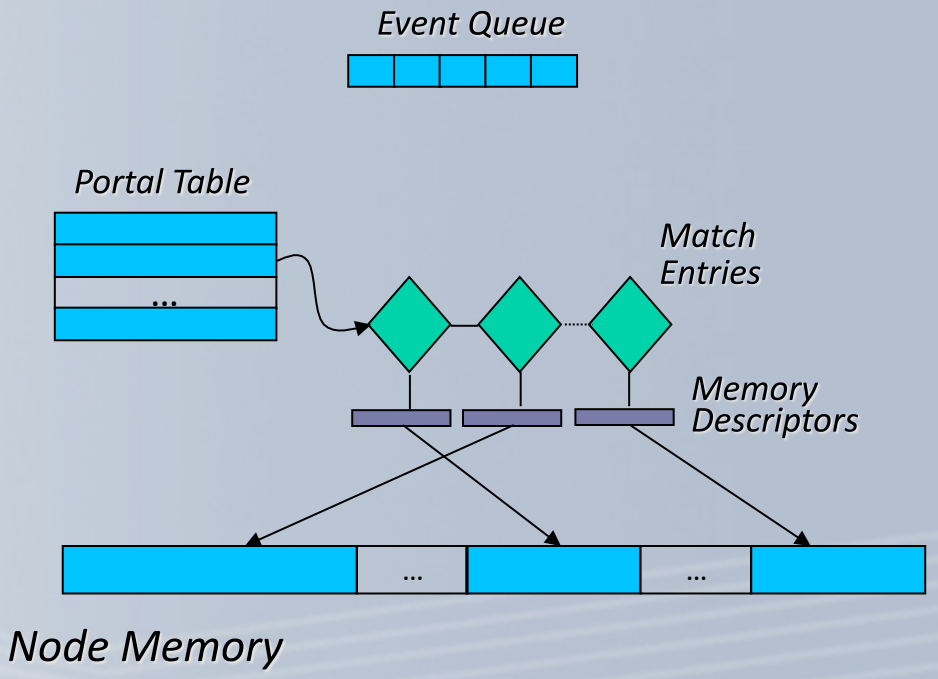
- Portal Table
  - Anchor table for registered portals users
- Event Queue
  - Status notification
- Match Entries
  - Filters for in-bound RDMA operations
- Memory Descriptors
  - Describe memory segments attached to match entries



CRAY
THE SUPERCOMPUTER COMPANY

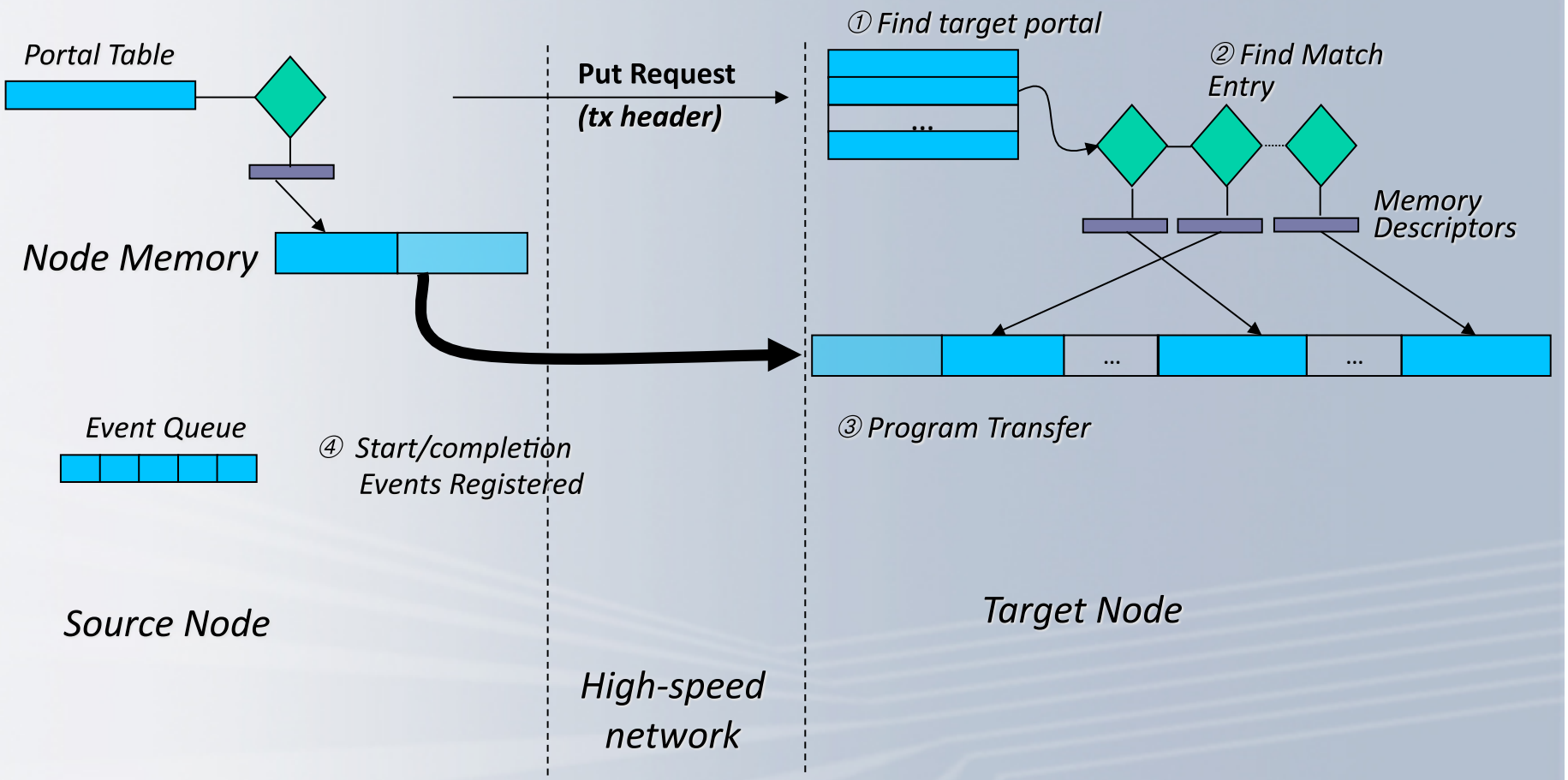
Portals API

- Portal Table
  - PtlNIInit(), PtlFini()
- Event Queue
  - PtlEQAlloc()
- Match Entries
  - PtlMEAttach()
- Memory Descriptors
  - PtlMDAttach(), PtlMDBind()
- Data Movement
  - PtlGet(), PtlPut()



CRAY
THE SUPERCOMPUTER COMPANY

Portals Data Transfer

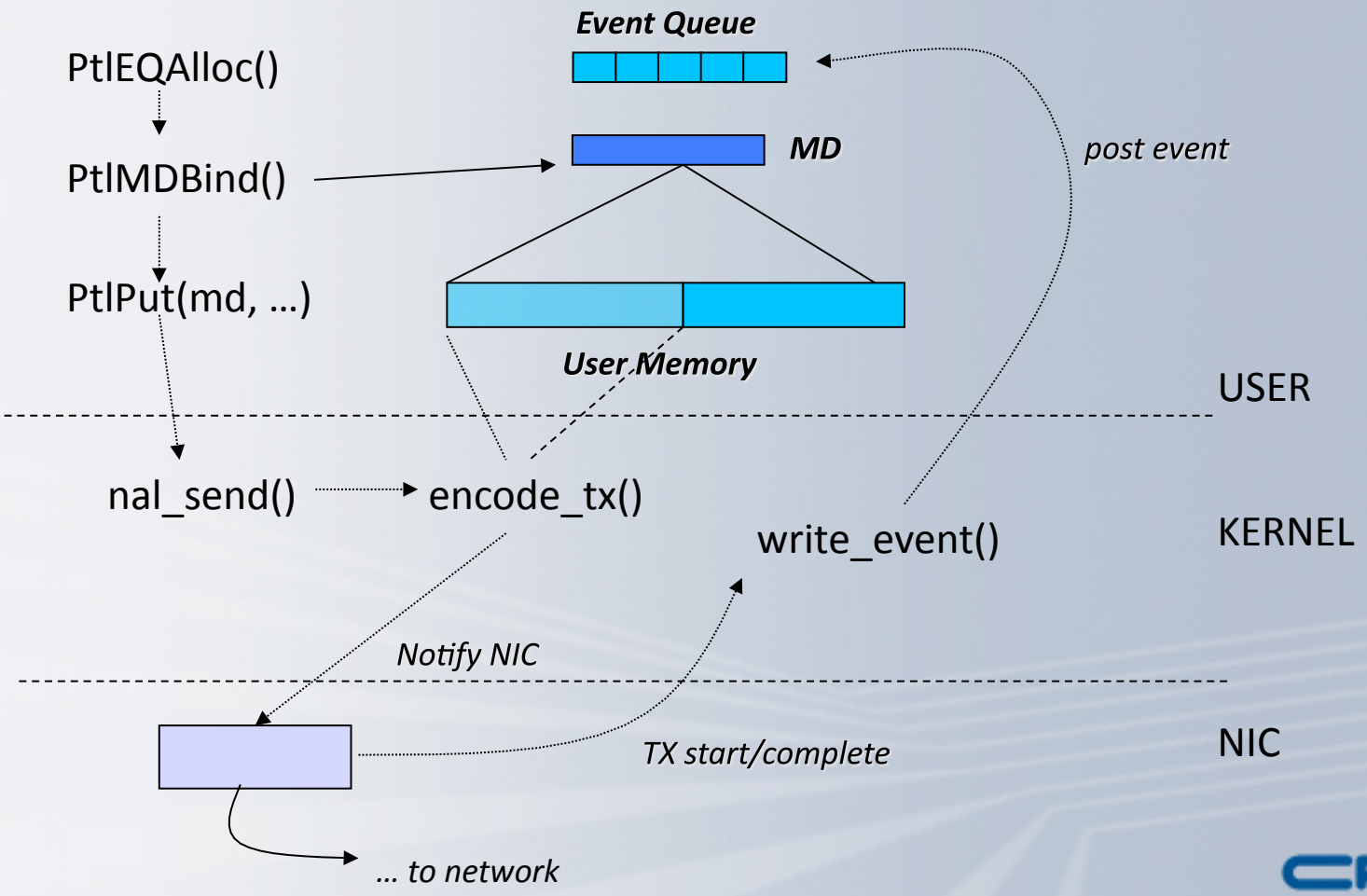


CRAY
THE SUPERCOMPUTER COMPANY

Portals Network Access Layer

- Provides bridge between portals API and NIC
- Manages transmit and receive state for in-flight portals requests
- Works in conjunction with portals library for message validation and matching
- Translates logical requests into DMA programs to be consumed by the NIC

User Transmit Request Flow through NAL

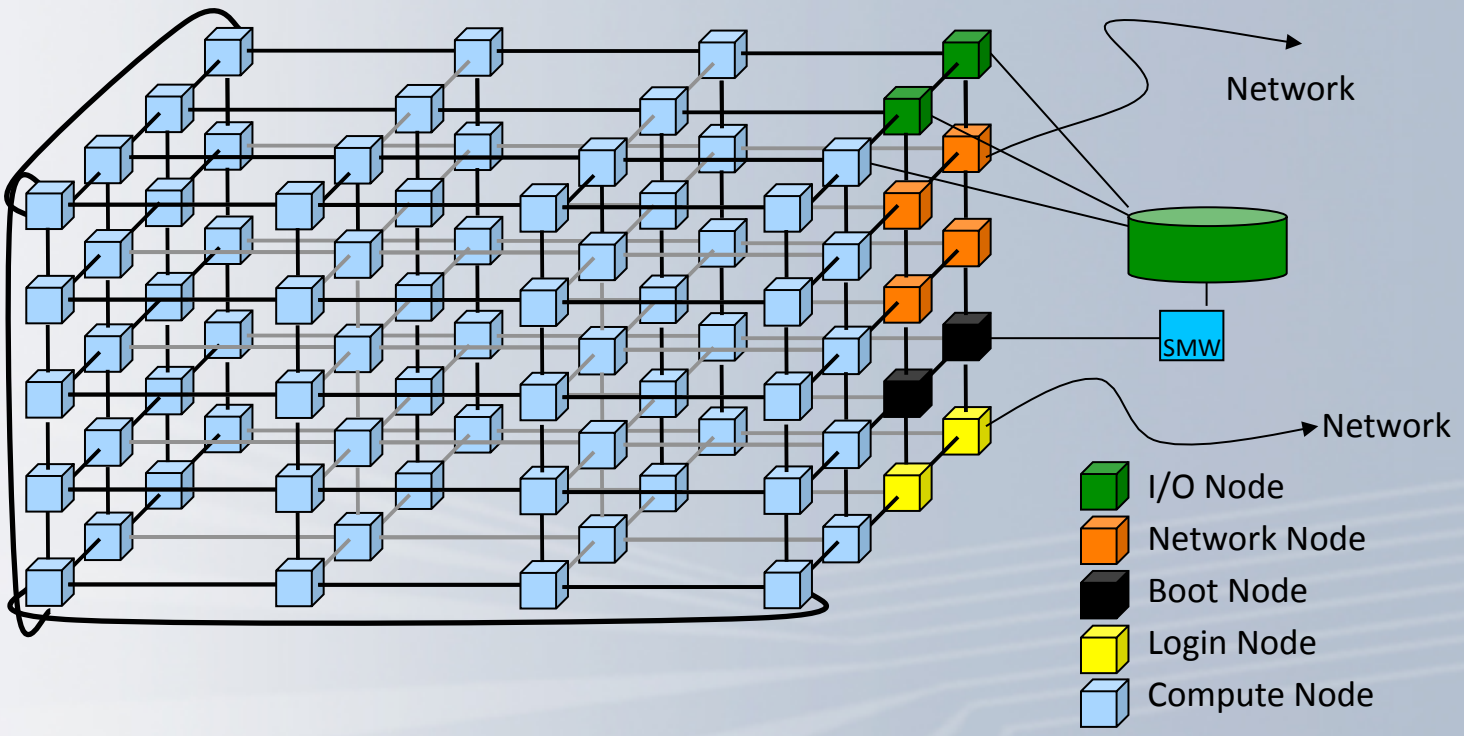


CRAY
THE SUPERCOMPUTER COMPANY

Providing reliable communication

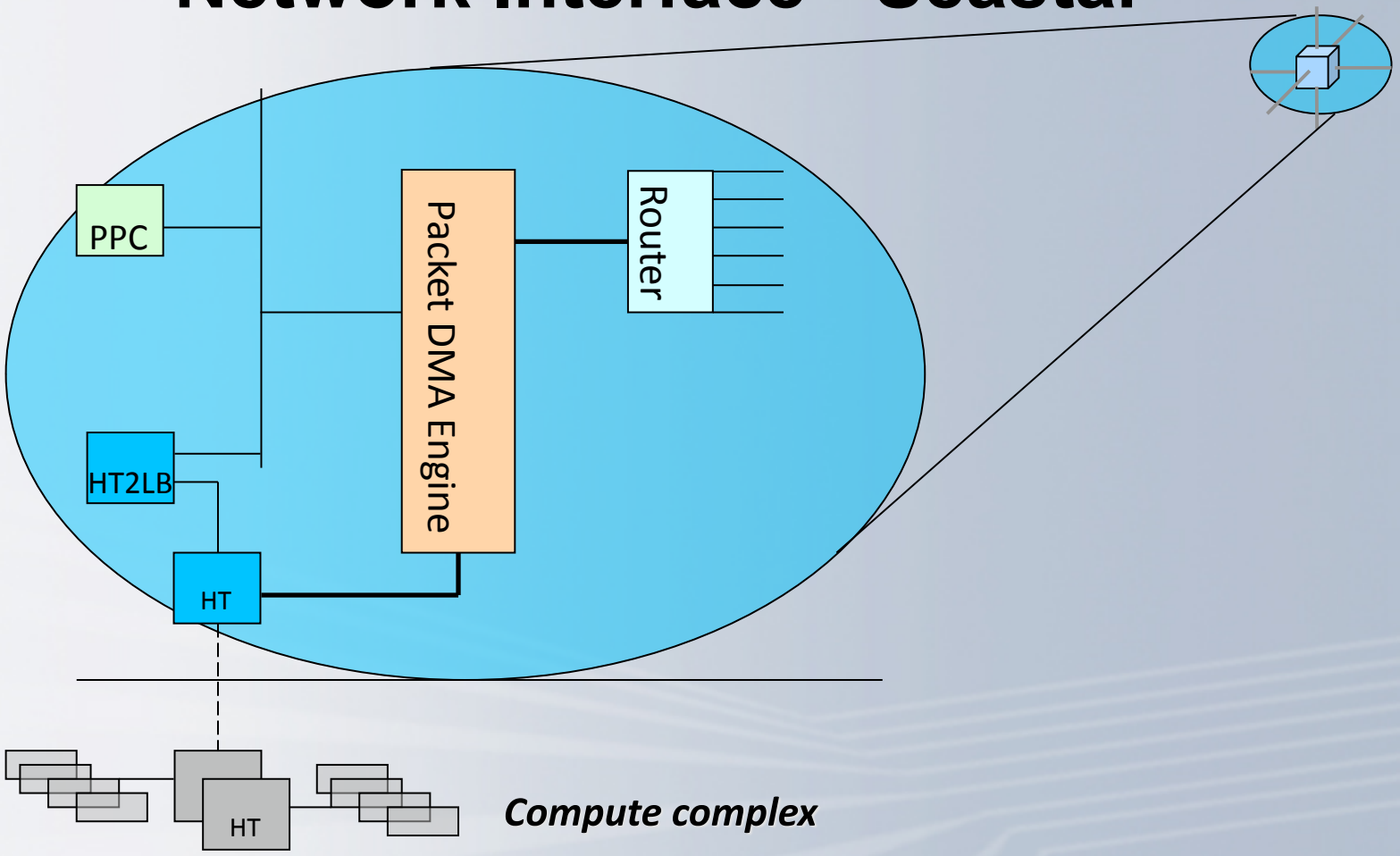
- Basic End-to-End Reliability (BEER) protocol
  - Initial goal was to provide CAM overflow recovery
- Host-resident software layer between NIC and portals
- Provides recovery for various message transmission issues
  - CAM overflow
  - OS resource exhaustion
  - Flow control for tx/rx imbalance and forward progress
  - Cleanup for orphaned transmits (remote node failures, etc.)
  - Warm reboot of nodes
  - Extended critical resource exhaustion handling (rx fifo full)
- Message sequence numbers used to manage flow between NID/PID pairs

XT System Architecture



CRAY
THE SUPERCOMPUTER COMPANY

Network Interface - Seastar



Compute complex

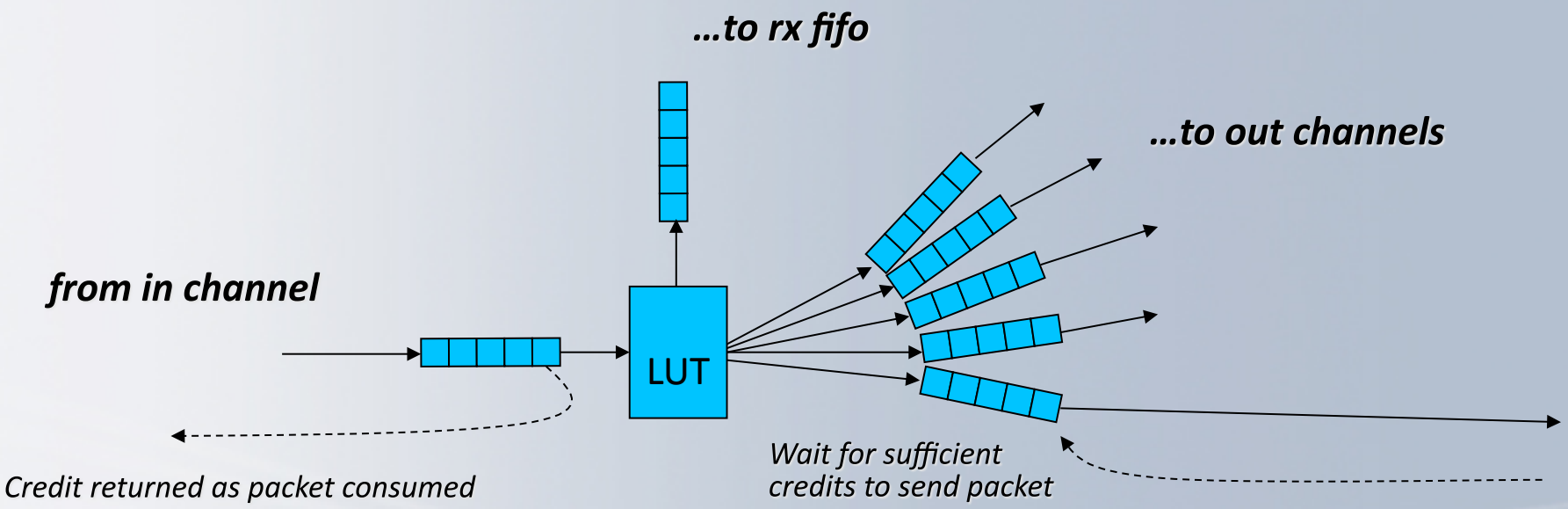
CRAY
THE SUPERCOMPUTER COMPANY

Seastar Router

- Connects the seastar to the 3D torus network
  - 6 network ports
  - 1 host port
- Supports up to 32K nodes
- Each port supports 3.25GB/s bandwidth
- Supports cut-through routing
- Basic flow control unit is a flit
  - Flit contains 64 data bits + control bits
- Flits are combined into network packets
  - Packets consist of a header flit and up to 8 data flits
- Hardware flow control is credit based
  - Credits are replenished when receiving router accepts flit

CRAY
THE SUPERCOMPUTER COMPANY

Seastar Router



LUT = Lookup Table: routes inbound request through switch to destination channel

CRAY
THE SUPERCOMPUTER COMPANY

Embedded Processor Complex

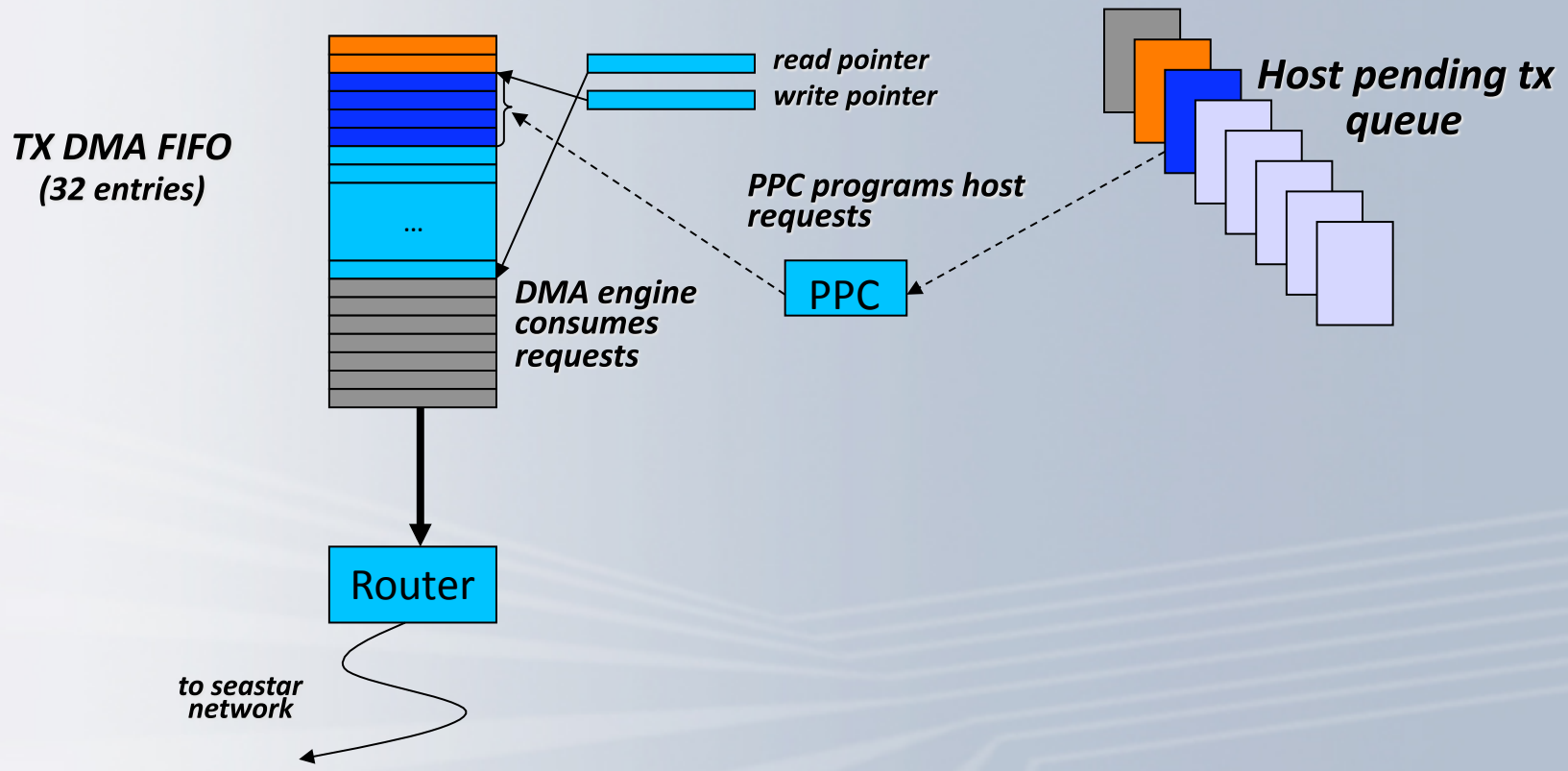
- PPC 440 Processor
- 384KB on-board RAM
- Manages DMA engine
  - Interrogates incoming requests and coordinates with host for data placement
  - Keeps tx engine fed
- Provides RAS functionality
  - Warm reboot
  - Host failure recovery
  - Maintenance interface communication
  - Seastar fault detection and recovery
  - Extended resource exhaustion detection

CRAY
THE SUPERCOMPUTER COMPANY

Seastar DMA Engine

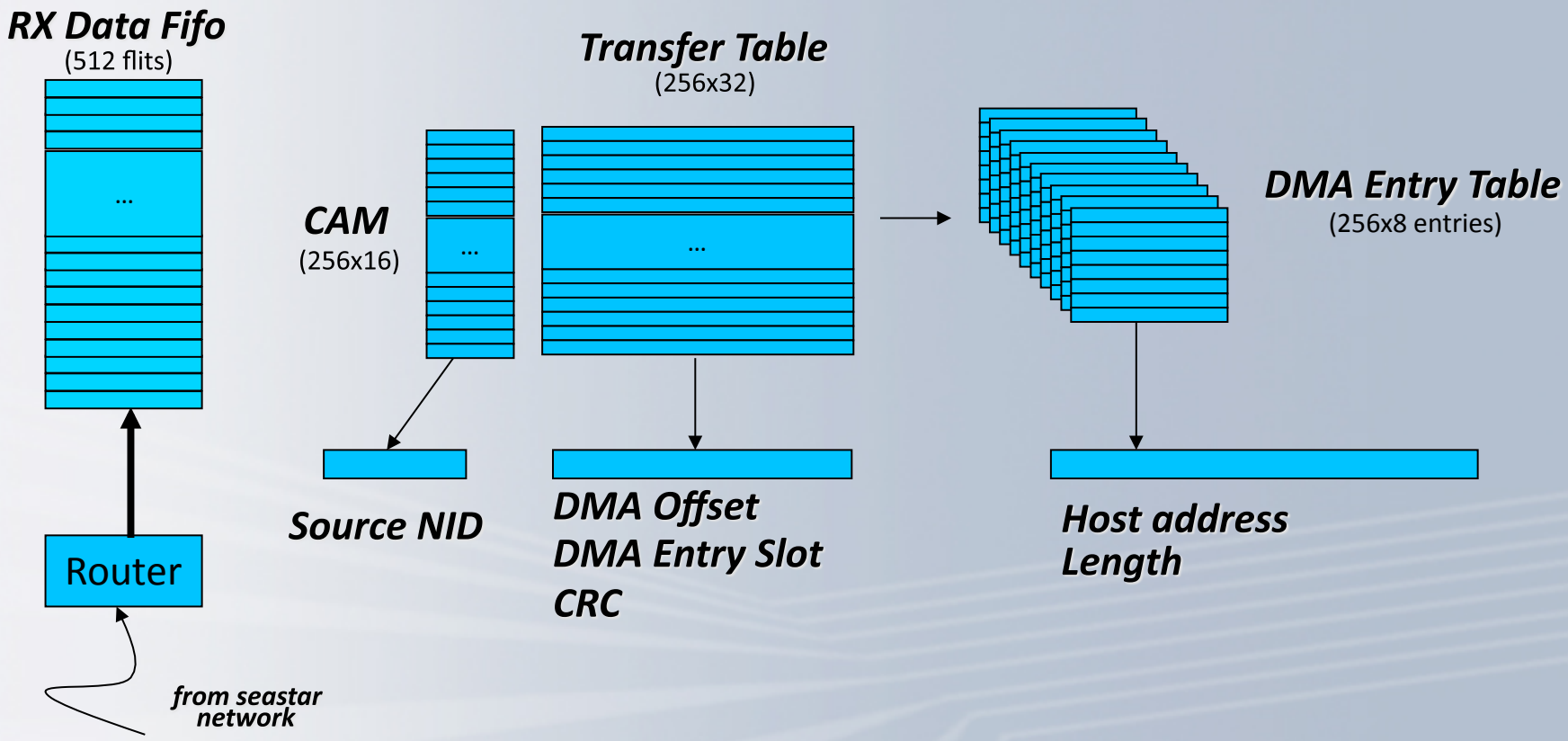
- Transmit Engine
  - Single transmit DMA engine
  - Responsible for translation of outbound block transfers into network flits
  - PPC manages 32 entry TX DMA queue
  - Price of tx = 1 credit / flit
- Receive Engine
  - Fed by 512 flit rx data fifo
  - Content Addressable Memory (CAM) maps remote source to each of the 256 rx queues
  - On CAM match, packet is sent to matching rx DMA stream
  - Router credited when packet is moved from rx data fifo

Seastar Transmit Machinery



CRAY
THE SUPERCOMPUTER COMPANY

Seastar Receive Machinery

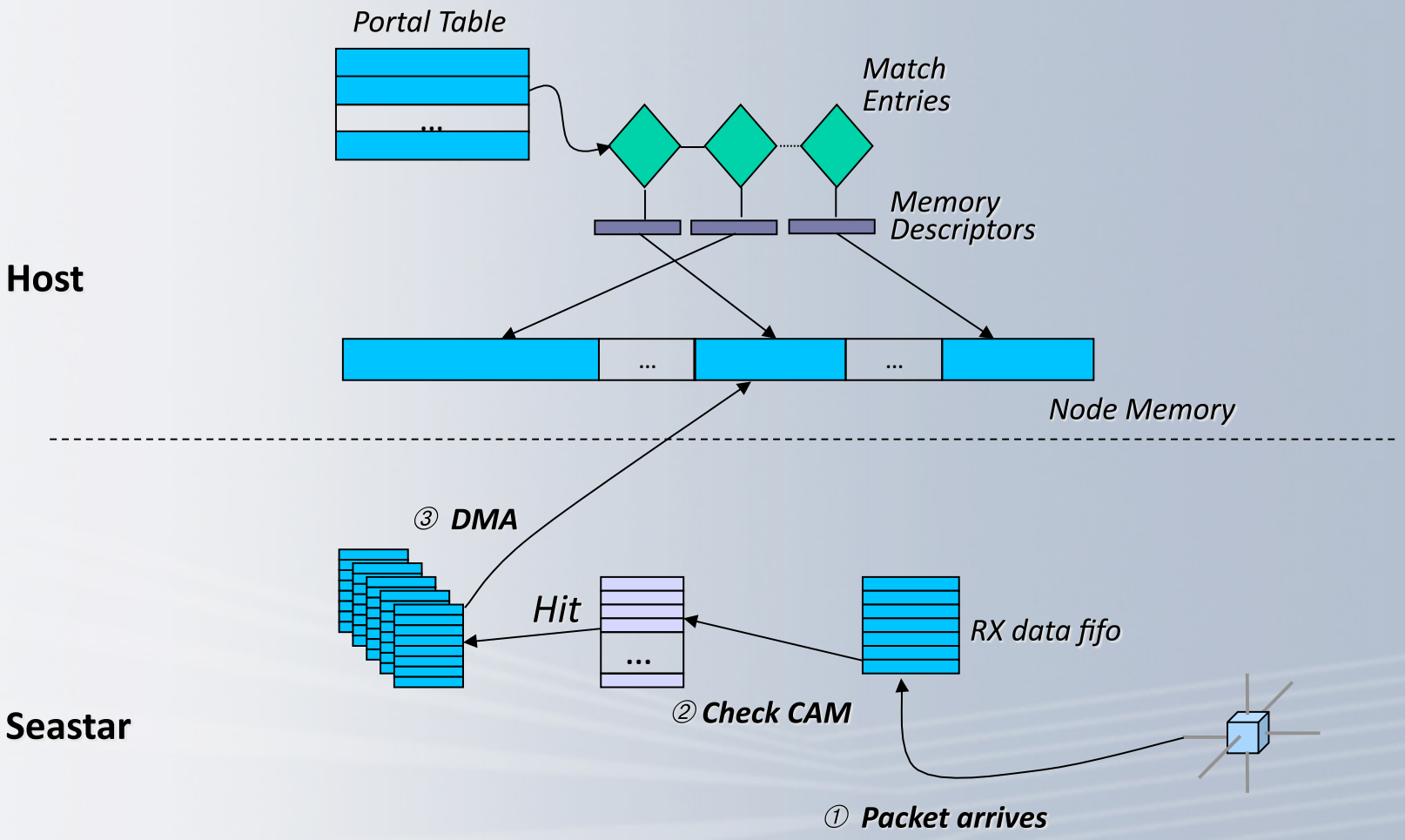


CRAY
THE SUPERCOMPUTER COMPANY

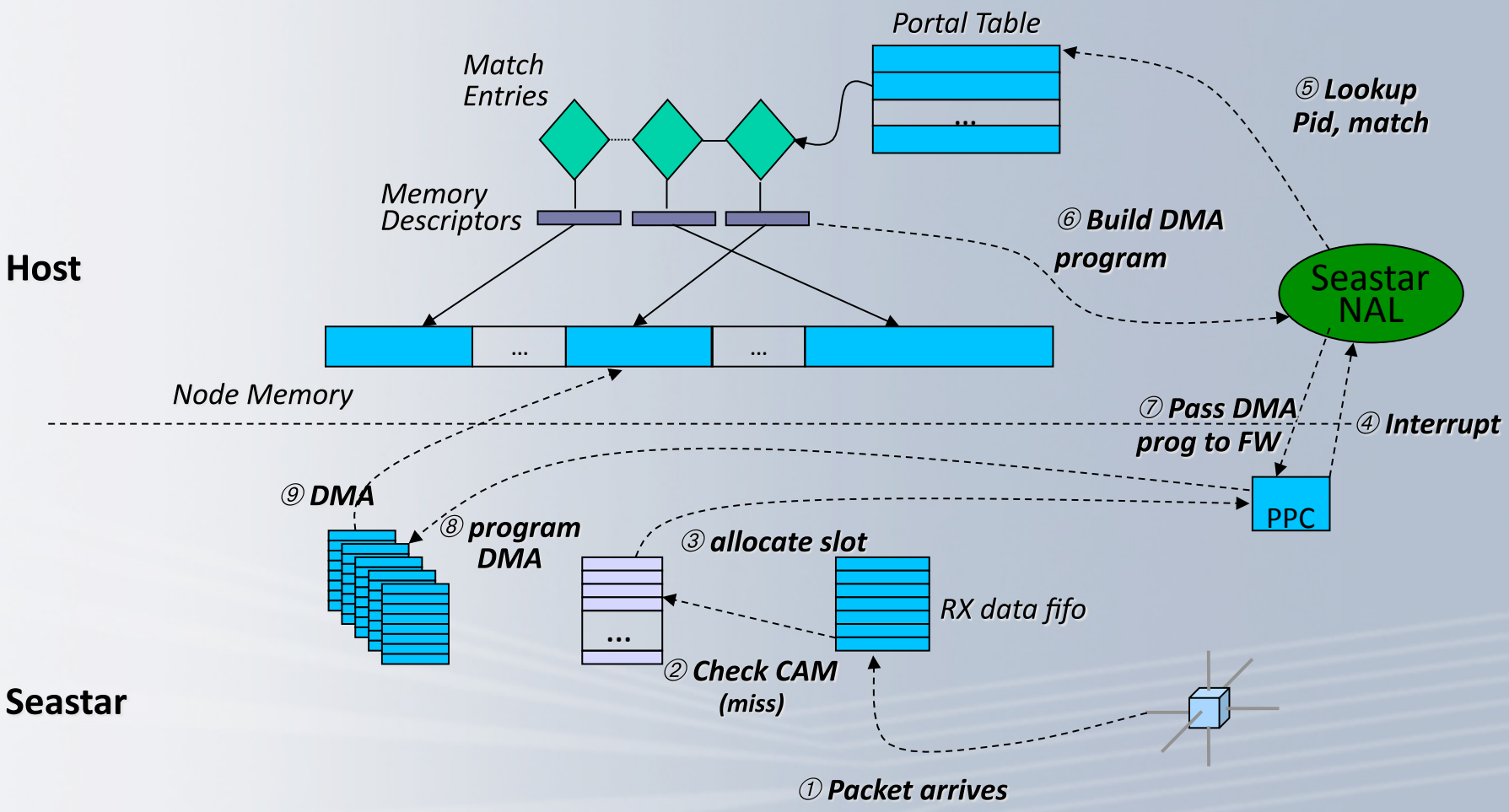
Key DMA Parameters

- TX DMA
  - Single transmit FIFO
  - Requests are consumed in the order presented
  - PPC responsible for feeding host requests into engine
  - Transmit possible only when credits available at target router
- RX DMA
  - Single receive FIFO
  - Small messages (<= 16 bytes) handled directly from FIFO
  - Large messages require CAM for DMA
  - PPC/host intervene on CAM miss
  - Router credits returned when packet moves from FIFO

Normal message reception

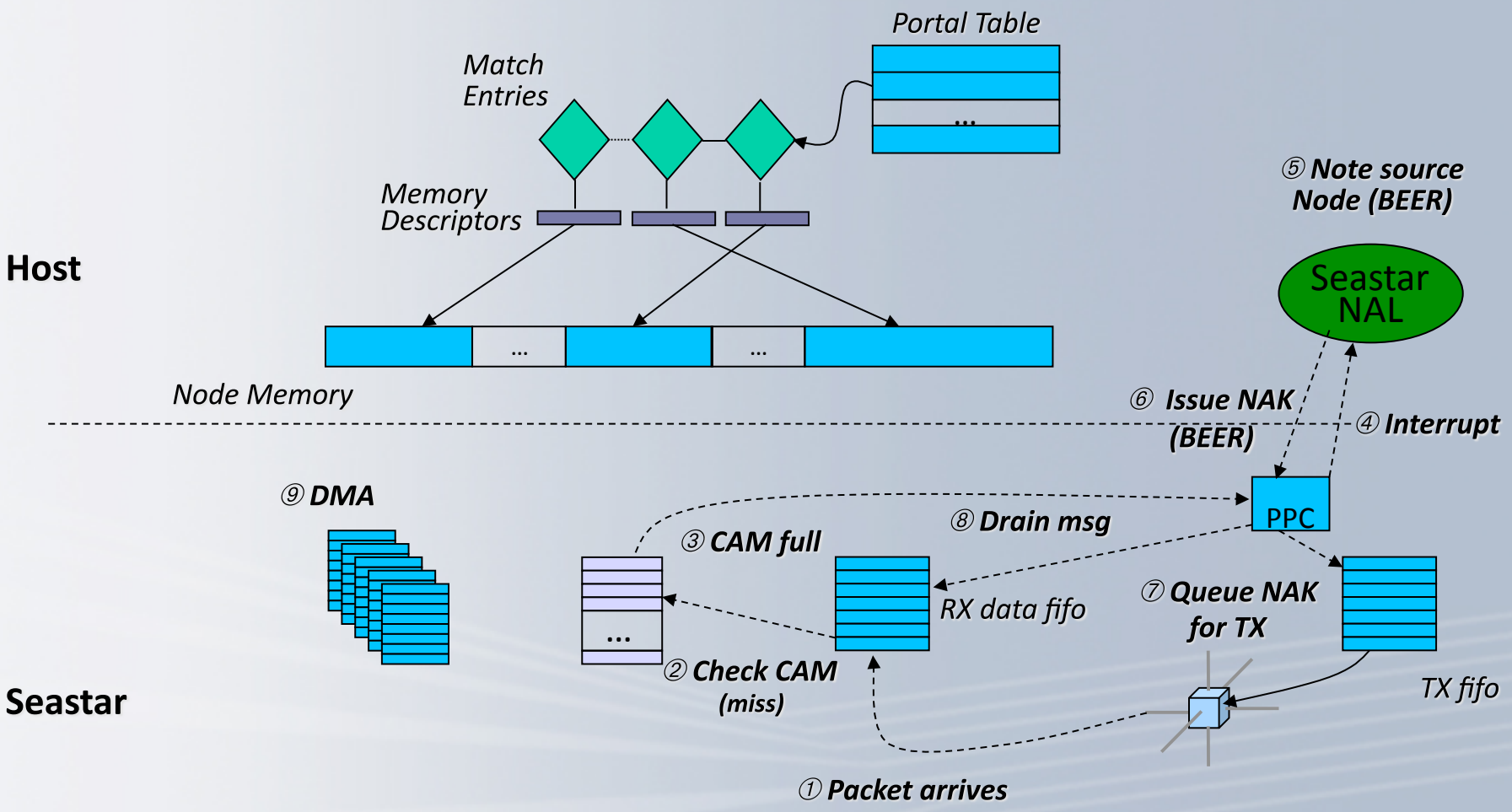


CAM Miss



CRAY THE SUPERCOMPUTER COMPANY

CAM Overflow



CRAY THE SUPERCOMPUTER COMPANY

Problem Scenario 1 – CAM thrash

- All-to-one
  - Actually, > 256 to one
- Effective throughput of the seastar decreases in direct proportion to the sender/receiver ratio
  - CAM overflows induce additional chatter due to:
    - Interrupt/nak sequence
    - Message retransmission
    - Unused data flowing through rx data fifo
- No fairness in CAM allocation scheme
  - CAM slots are ‘sticky’, but can be resourced at the end of a message.

Scenario 2 – tx/rx imbalance

- All-from-one (e.g., GET storm)
  - Unrelated to CAM issues
- GET requests arrive as single-packet requests, which do not require the CAM
  - Processed directly from rx fifo
  - Not throttled by CAM limit of 256 concurrent sources
- REPLY DMAs fill the tx fifo and requests begin to backup
- BEER initiates flow control when transmit resources become scarce

Scenario 3 – tx interference

- Transmit requests serialized through Seastar
- Large tx stream from one PE can monopolize resources
- Transmit queue is not prioritized
  - Don’t initiate many large low-priority (async) requests followed by a small, high priority request (e.g., barrier)
- Must consider asynchronous tx activity
  - Lustre buffer flushes
  - Out-of-band requests (RDMA pulls)

Summary

- Forward progress on the Seastar network is key
  - Stalls/delays at one node can ripple through the network.
- All-to-one communication comes at a price
  - CAM pressure can reduce effective throughput
- Understand unique characteristics of tx and rx communication paths
- Out-of-band communication can have significant impact on application
- Be mindful of resource exhaustion costs as application scales